

Angular Diciannove, Quattro.

Forms

Template Driven Forms

Reactive Forms

Kas yra forma?

HTML forma - standartinis būdas surinkti duomenis iš vartotojo.

```
<form>  
  <label for="text">Text</label>  
  <input id="text" type="text" />  
</form>
```

Kas yra form controls

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input#input_types

- Form controls yra pagrindiniai formos elementai, leidžiantys vartotojui įvesti duomenis.
- Įvairūs naršyklės *native input* tipai leidžia naudoti naršyklės *native* konfigūraciją ir *input* reikšmių validaciją.

```
<form>
  <ul>
    <li>
      <label for="text">Text</label>
      <input id="text" type="text" />
    </li>
    <li>
      <label for="number">Number</label>
      <input id="number" type="number" />
    </li>
    <li>
      <label for="date">Date</label>
      <input id="date" type="date" />
    </li>
    <li>
      <label for="password">Password</label>
      <input id="password" type="password" />
    </li>
    <li>
      <label for="color">Color</label>
      <input id="color" type="color" />
    </li>
    <li>
      <label for="range">Range</label>
      <input id="range" type="range" />
    </li>
  </ul>
</form>
```

CONTACT

Name:

E-mail:

Message:

SEND YOUR MESSAGE

Contact

- Name
- E-Mail
- Message

Send Your Message

```

<form name="messageForm" action="/message" method="POST">
  <fieldset>
    <legend>Contact</legend>
    <ul>
      <li>
        <label for="name">Name</label>
        <input name="name" type="text" id="name" />
      </li>
      <li>
        <label for="email">E-Mail</label>
        <input name="email" type="email" id="email" />
      </li>
      <li>
        <label for="message">Message</label>
        <textarea name="message" id="message"></textarea>
      </li>
    </ul>
    <button>Send Your Message</button>
  </fieldset>
</form>

```

Contact

- Name
- E-Mail
- Message

Send Your Message

Angular forms.

Template driven ir Reactive

- **Template driven forms**

- Naudoja ***two-way data-binding***, kad atspindėtų duomenų pokyčius tarp template ir kodo. Validacija atliekama naudojant **standartinius *HTML*** validavimo mechanizmus.
- Puikiai tinka paprastoms, greitoms formoms.

- **Reactive forms**

- Naudoja ***immutable*** duomenų modelį, kuris yra nepriklausomas nuo template ir kiekvieną kartą unikalus.
- Tinka **sudėtingoms, duomenų gausioms formoms**, kur reikia ***cross-field*** arba **asinchroninės** validacijos. Galima testuoti formos logiką *unit test'ais*.

<https://angular.dev/guide/forms/template-driven-forms>

Template driven forms

```
<form #expenseForm (ngSubmit)="addExpense()">
  <div class="row">
    <div class="col">
      <input
        type="text"
        class="form-control"
        placeholder="Expense name"
        name="name"
        [(ngModel)]="model.name"
      />
    </div>
  </div>
</form>
```

```
@Component({
  selector: 'app-contact',
  imports: [FormsModule],
  templateUrl: './contact.component.html',
  styleUrls: ['./contact.component.css']
})
export class ContactComponent {}
```

Template driven forms

- Importuojama iš *FormsModule*.
- Template pateikiamas formos **modelis**, kur kiekvienas form control atitinka tam tikrą modelio savybę.
- Leidžia susieti skirtingus form control, naudojant *ngForm* direktyvą.

```
@Component({  
  selector: 'app-contact',  
  imports: [FormsModule],  
  templateUrl: './contact.component.html',  
  styleUrls: ['./contact.component.css']  
})  
export class ContactComponent {}
```

```
export class Message {  
  constructor(  
    public name: string,  
    public email: string,  
    public text: string) {}  
}
```


ngModel ir two way data binding

- Naudoja ***two-way-binding*** sintaksę: reikšmei nurodyti – **[]**, o pokyčiams klausyti – **()**. Visa tai kartu sudaro ***[(ngModel)]***.
- Tai yra ***[(ngModel)]*** direktyvos ir ***(ngModelChange)*** event kombinacija.

```
<input
  type="text"
  name="message"
  [value]="message"
  (input)="message = $any($event).target.value"
/>
<input
  type="text"
  name="message"
  [(ngModel)]="message"
  (ngModelChange)="message = $event"
/>
<input type="text" name="message" [(ngModel)]="message" />
```

Template modelis, formos modelis ir template kintamieji.

- **#name** aprašo template kintamuosius Angular'e – tai kintamieji, kurie pasiekiami tik HTML template.
- Pateikia **ngForm** kaip **message** kintamąjį šablone.

```
<form #messageForm="ngForm" (ngSubmit)="sendMessage()">
```

- Pateikia **ngModel** kaip **message** variable šablone.

```
<input type="text" name="message" [(ngModel)]="message" #message="ngModel"/>
```

- „id“ naudojamas **labels**, o „name“ – form control susiejimui su forma.

Formos submit

```
<button>Send Your Message</button>
```

```
<form #messageForm="ngForm" (ngSubmit)="sendMessage()">
```

```
sendMessage() {  
    this.#service.send(this.model);  
}
```

```
<button (click)="sendMessage()">Send Your Message</button>
```

Template validacijos

- Naudoja **HTML** validacijos atributus.
- **Required, minlength, maxlength, pattern, t.t.**

```
<input  
  name="name"  
  type="text"  
  id="name"  
  [(ngModel)]="model.name"  
  required  
  placeholder="Your name"  
>
```

- Galima naudoti template kintamuosius, norint pasiekti validacijos parametrus:

```
<button (click)="sendMessage()" [disabled]="!messageForm.valid">  
  Send Your Message  
</button>
```

Template Driven State in Stiliai

State	Class if true	Class if false
The control has been visited.	ng-touched	ng-untouched
The control's value has changed.	ng-dirty	ng-pristine
The control's value is valid.	ng-valid	ng-invalid

```
.ng-invalid:not(form) {  
    border-left: 5px solid red;  
}
```

```
<label for="name">Name</label>  
<input  
  name="name"  
  type="text"  
  id="name"  
  [(ngModel)]="model.name"  
  required  
  placeholder="Your name"  
  #messageName="ngModel"  
>  
</li>  
<div [hidden]="messageName.valid || messageName.pristine">  
  @if(messageName.errors?.['required']) {  
    Message name is required  
  }  
</div>
```

Task #1 - Template Driven Forms

1. Pridėk formą prie **TasksComponent**, skirtą naujų užduočių pridėjimui. Forma turi turėti du laukus: „text“ (text input) ir „date“ (date input).
2. Naudokite šiuos įvesties laukus vietoje statinių teksto / datos reikšmių užduoties pridėjimo metu.
3. Abu laukai – „text“ ir „date“ – yra privalomi, o „date“ turi leisti pasirinkti tik datas po šiandienos.
4. Jei nėra įvestas tekstas, turi būti parodyta klaida.